

Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1-10. (Cancelled)

11. (Currently Amended) A machine-implemented method comprising:

generating parallel processes in a data processing machine; effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes;

wherein the one or more instructions comprise an instruction including an operand indicating an address from which to start executing instructions if a mis-speculation occurs.

12. (Original) The method of claim 11, wherein said generating parallel processes comprises running a software

program that spawns multiple threads in the data processing machine.

13. (Original) The method of claim 11, wherein said effecting synchronization comprises translating at least one high-level software instruction into at least one machine instruction that controls speculative execution in a processor.

14. (Original) The method of claim 11, wherein said effecting synchronization comprises placing in an out-of-order execution management unit of a processor, at least one machine instruction that limits when other machine instructions are retired from the out-of-order execution management unit.

15. (Original) The method of claim 11, wherein said providing output comprises sending the output to another data processing machine.

16. (Currently Amended) An article comprising a machine-readable storage medium embodying information indicative of instructions that when performed by one or more machines result in operations comprising:

generating parallel processes in a data processing machine;

effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes;

wherein the one or more instructions comprise an instruction including an operand indicating an address from which to start executing instructions if a mis-speculation occurs.

17. (Original) The article of claim 16, wherein said generating parallel processes comprises running a software program that spawns multiple threads in the data processing machine.

18. (Original) The article of claim 16, wherein said effecting synchronization comprises translating at least one high-level software instruction into at least one machine instruction that controls speculative execution in a processor.

19. (Original) The article of claim 16, wherein said effecting synchronization comprises placing in an out-of-order

execution management unit of a processor, at least one machine instruction that limits when other machine instructions are retired from the out-of-order execution management unit.

20. (Original) The article of claim 16, wherein said providing output comprises sending the output to another data processing machine.

21. (Currently Amended) A machine-implemented method comprising:

speculatively executing machine instructions, including a memory access instruction, in a processing system to effect synchronization between parallel processes, wherein the speculatively executing comprises performing a speculative read-modify-write to a lock variable associated with a critical section;

retiring the speculatively executed machine instructions to end speculation before performing the critical section; and

maintaining cache coherence in the processing system during said executing and said retiring to identify a mis-speculation to effect the synchronization between the parallel processes;

wherein the machine instructions comprise an instruction including an operand indicating an address from which to start executing instructions if the mis-speculation occurs.

22. (Original) The method of claim 21, wherein said maintaining cache coherence comprises providing invalidation based cache coherence.

23. (Original) The method of claim 21, wherein said speculatively executing machine instructions comprises speculatively executing machine instructions in the processing system comprising multiple processors, and the mis-speculation comprises a memory dependency violation.

24. (Original) The method of claim 21, wherein the mis-speculation comprises at least one of an interrupt, an external event, and a memory dependency violation.

25. (Currently Amended) A system comprising:  
a processor having a processor architecture that provides speculative execution of machine instructions and exposes said speculative execution to program control through at least one machine instruction; and  
a memory coupled with the processor, the memory embodying information indicative of instructions, including the at least one machine instruction, that result in synchronization between

parallel processes when performed by the processor with  
detection of mis-speculation;

wherein performance of the instructions by the processor  
comprises performing a speculative read-modify-write to a lock  
variable associated with a critical section and ending  
speculation before performing the critical section;

wherein the at least one machine instruction comprises:  
a speculative execution instruction that takes first and  
second operands, behaves as a no-op if a memory location  
indicated by the first operand contains a first value, causes  
the processor to speculatively execute additional instructions  
if the memory location contains a second value, and causes the  
processor to start executing instructions from an address  
indicated by the second operand if a mis-speculation occurs; and  
a speculation termination instruction that causes the  
processor to begin retiring the additional instructions if the  
additional instructions have been speculatively executed.

26. (Original) The system of claim 25, wherein the  
processor comprises a uniprocessor.

27. (Original) The system of claim 25, wherein the  
processor comprises a multiprocessor.

28. (Original) The system of claim 27, wherein the multiprocessor comprises multiple processing units on a single die.

29. (Original) The system of claim 25, further comprising: a communication interface; and a virtual machine that translates the information, received via the communication interface, into the at least one machine instruction.

30. (Original) The system of claim 29, wherein the virtual machine comprises a Java virtual machine.

31. (Cancelled)

32. (Original) The system of claim 25, further comprising an environmental sensor coupled with the processor.

33. (Currently Amended) A processing system comprising: processing means for speculatively executing machine instructions in response to a speculative execution instruction, including means for detecting a mis-speculation; means for treating multiple speculative instructions as a group for purposes of retirement such that the multiple

speculative instructions are flushed from the processing means together and execution proceeds from an address in response to a detected mis-speculation to effect synchronization between parallel processes;

wherein performance of the instructions by the processing means comprises performing a speculative read-modify-write to a lock variable associated with a critical section and ending speculation before performing the critical section;

wherein the speculative execution instruction includes an operand indicating the address.

34. (Original) The processing system of claim 33, wherein said means for detecting a mis-speculation comprises means for maintaining cache coherence in the processing means.